Массив

**Массив** – это объект, в котором ключом является индекс элемента массива, а значением – элемент массива.

Определение массива:

const arr = [];

typeof ar; //object

[] – **Синтаксический сахар для** new Array()

Delete arr[1] – удаляет значение элемента с индексом 1 в объекте типа Array, но не сам элемент этого объекта. Удаление элемента производят созданием нового массива через фильтрацию исходного.

Итерацию по массиву лучшу проделывать не циклом, а функциями высшего порядка **map, filter, redkuce**

Запись **arr.map(Math.sqrt)** аналогична **arr.map(value => Math.sqrt(value))**

Итерация обычного объекта происходит через получение массивов ключей или значений через конструктор **Object**:

**Object.keys(obj).forEach(key => console.log(obj[key]));**

**Object.values(obj). forEach(value => console.log(value));**

**Object.keys(obj).length** // определение размера объекта

**Ключи у объекта только строки и символы**.

Ассоциативный массив

**Ассоциативный массив** – массив из пар **ключ-значение**

map = new Map([["key1", "value1"], ["key2", "value2"]]);

map.set("key3", "value3”); // добавление пары в массив

map.get("key1"); // value1

map.keys(); // { 'key1', 'key2' }

map.values(); // { 'value1', 'value2' }

map.entries(); // { [ 'key1', 'value1' ], [ 'key2', 'value2' ] }

map.forEach((value, key) => console.log(key, value)); // key1 value1 / key2 value2 / undefined

map.size // 2 (размер Map)

Множества

Объекты **Set** представляют коллекции значений, по которым вы можете выполнить обход в порядке вставки элементов. Значение элемента в Set может присутствовать только в одном экземпляре, что обеспечивает его уникальность в рамках коллекции Set.

set = new Set(); // Set {}

set.add(5); // Set { 5 }

set.add('text'); // Set { 5, 'text' }

set.size // 2

set.has(5) // true

set = new Set([1, 2, 3, 4, 4, 5, 5]); // Set { 1, 2, 3, 4, 5 }

Array.from(set); // [ 1, 2, 3, 4, 5 ]

Rest

Синтаксис **rest** функции позволяет представлять неограниченное множество аргументов в виде массива.

function(a, b, ...theArgs) {

// …

}

Если последний именованный аргумент функции имеет префикс ..., то он автоматически становится массивом с элементами от 0 до theArgs.length в соответствии с актуальным количеством аргументов, переданных в функцию.

Spread

Оператор **spread** позволяет расширять выражения в тех местах, где предусмотрено использование нескольких аргументов (при вызовах функции) или ожидается несколько элементов (для массивов).

При вызовах функции:

f(...iterableObj);

В литералах массива:

[...iterableObj, 4, 5, 6]

При деструктивном присваивании:

[a, b, ...iterableObj] = [1, 2, 3, 4, 5];